

University of Malaya  
Faculty of Computer Science and Information Technology  
Department of Software Engineering

**Data Structure**   
**WIA1002**

Project Report  
**Project Title**: “Hospital Queuing System”

**Team name: Group 5  
Names and Matrix numbers:**  
1. Iffat Ahmed Afia WIC180705  
2. Asiri Ziad Ahmed: WID180712  
3. Danish Izzat Ang bin Yusri Danial Ang: WIG180007  
4. Shahan Ali Pranto: WIF180719  
6. Faiyaz Khan: WIF180720

Tutorial Group: 1

**Introduction**

We were required to simulate a hospital queuing system which has two parts- patients queuing up and doctor treating patients. We have used “Queue” as our data structure that has been coded from scratch. No data structure from the library were imported to be used in our project.

**Extra Methods**

Although we have used two extra methods which are “displayBothQinfo()” and areBothQempty(),other methods of our queue class were modified according to our needs.

**displayBothQinfo()**

This method shows the information of every patients enqueued in the queue after the enqueuing is finished. At first it shows the information by gender, then by age, and lastly by illness. All information is generated randomly.



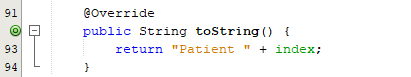
**areBothQempty()**

It checks if the queues are empty or not. If the queues are empty, it returns “true”. Otherwise it returns “false”.



**Overridden Methods**

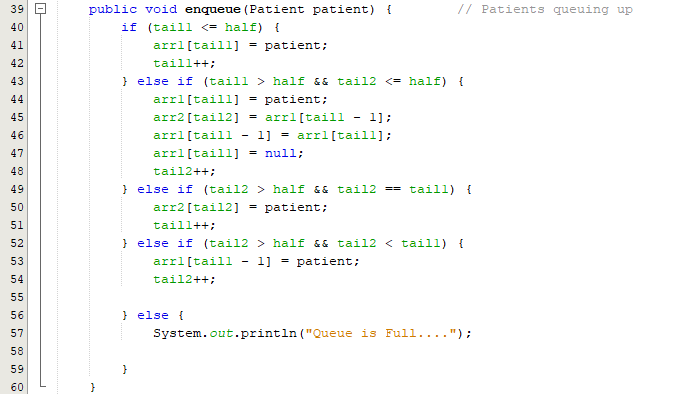
**toString()**

****

**Explanation of the algorithm**

**Patients queuing up**

The enqueue() is a method in class “Queue” which enqueue patient objects in two arrays. The two arrays have two pointers called “tail1” and “tail2”. The pointers start from 0. The “maxSize” is the number of patients in each queue.

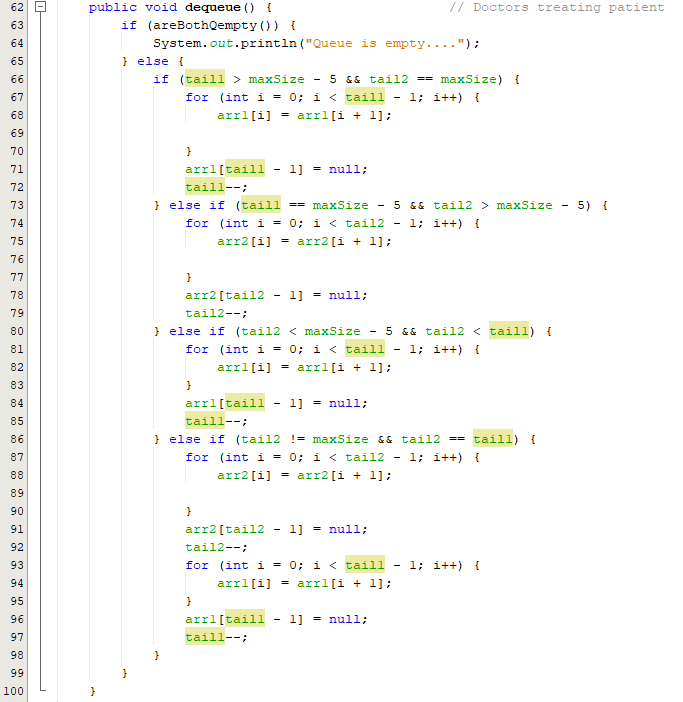
When the number of patients in the first array is less than or equal to half the “maxSize”, the patients are enqueued in the first array and the pointer of that array is incremented. When the number of patients in the first array gets more than half the “maxSize”, the next patient is enqueued in the first array, the second last patient of the first array gets shifted to the second array. The last patient of the first array moves up one space and the pointer of first array points to null. The second array pointer is incremented. This continues until both the arrays are half full. Afterwards, when both the pointers point at the same level of their respective arrays, the next patient is enqueued in the second array and the pointer in first array is incremented. Then the next patient lines up in first array and the pointer in second array is incremented. This goes on until both the arrays reach “maxSize”.

**Doctors treating patients**

The dequeue() method of the class queue deques according to the algorithm given in the question. It removes patient object from array1 if “tail1 > maxSize-5” and “tail2 == maxSize”. This is because, according to the question, doctor 1 will first treat the first 5 patients of the queue 1. ”maxSize” is the size of one queue. The condition also indicates that the loop will keep removing objects until 5 objects have been removed from array1.

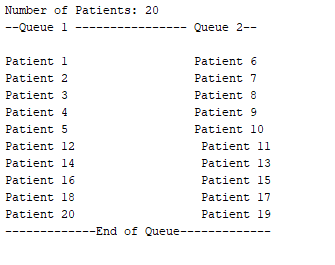
If “tail1 == maxSize-5 and tail2 > maxSize – 5”, the method will remove patient object from array2. This is because, according to the question, doctor 2 will treat the first 5 patients of queue 2 after doctor 1 has finished treating the first 5 patients of queue 1. The condition indicates that the loop will keep removing objects until 5 objects have been removed from array2.

Lastly when these two parts are done, doctors treat their patients at the same time until there are no more patients left. Therefore, we used the condition “tail2 != maxSize and tail2 == tail1”. As a result, after 5 objects from array 1 and array2 have been removed, this loop removes the objects from array 1 and array 2 simultaneously until there are no more objects left.



**Sample Outputs**

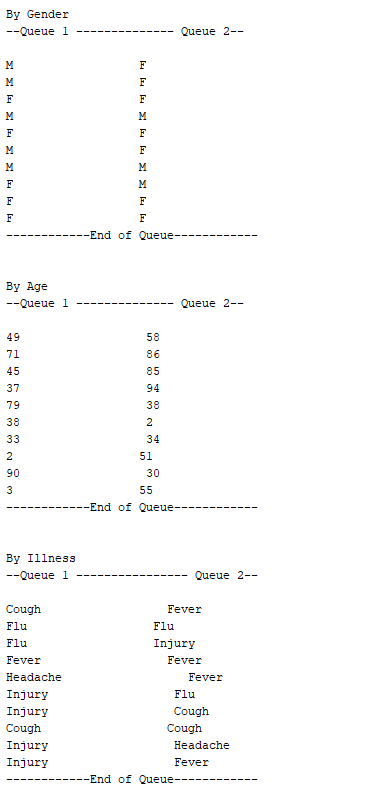
Patients enqueued and displaying patients



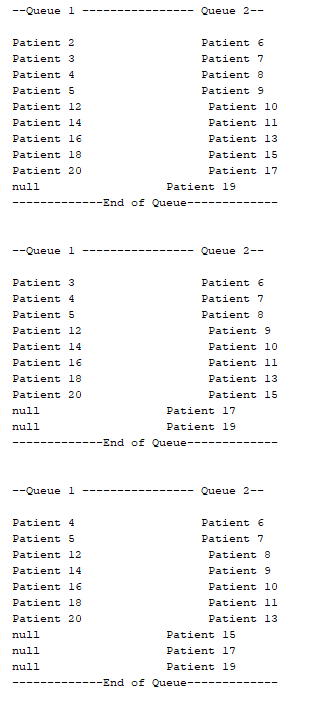
Checking if queues are empty



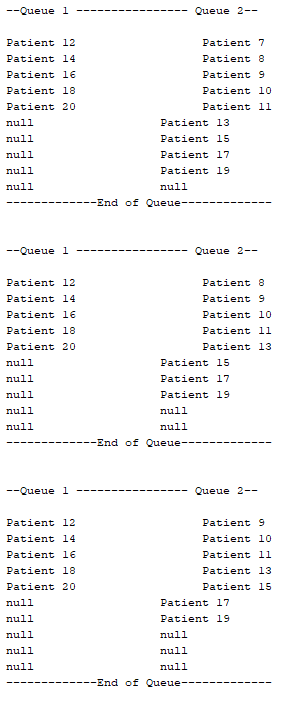
Displaying patients’ information



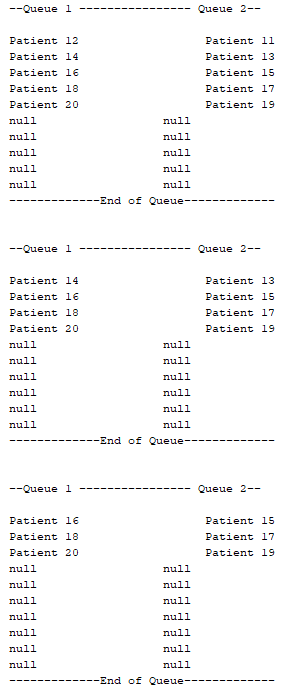
Doctors treating patients(dequeuing)



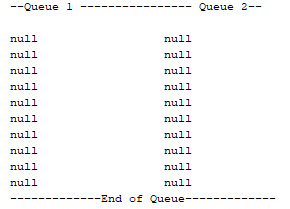
Skipping……..



Skipping……..



Skipping……..



Checking again if queues are empty



Note: Our system is not limited with 20 patients, but works as well with 10, 30, 100, 500….. patients.

Demo:

